Faculdade de Engenharia da
Universidade do Porto
Licenciatura Informática e Computação

Laboratório de Informática Avançada

Automatização de Horários

Relatório de Especificação de Requisitos

Versão 1.1

João Braga
http://www.fe.up.pt/~ei97027/lia.html
ei97027@fe.up.pt

Teresa Ribeiro
http://www.fe.up.pt/~ei97013/lia.html
ei97013@fe.up.pt

Porto, 25 de Maio 2001

Índice

	Pág.
1.Introdução	**1**
2.Lista de requisitos do sistema	**1**
3.Modelo de Casos de Uso	**3**
3.1 Visão Geral do Sistema	**4**
3.2 Actores do Sistema	**5**
3.3 Casos de Uso Exteriores ao Sistema	**6**
3.4 Casos de Uso Interiores ao Sistema	**13**
3.5 Exemplos de Interfaces	**16**
4. Requisitos Suplementares	**16**
5. Modelo de objectos do domínio	**18**
6. Algoritmo para a Geração de Horários 1	**19**
7. Algoritmo para a Geração de Horários 2	**19**
8 Integração com sistemas existentes	**20**
9 Validações junto do Cliente	**20**
10 Glossário	**21**

Anexos

Diagrama de Classes em UML

Fluxograma para a Geração de Horários

1. Introdução

O objectivo deste projecto é desenvolver uma aplicação que permita efectuar a geração semi-automática de horários, utilizando para tal um sistema Multi-Agente.

Este tipo de Arquitectura pressupõe o desenvolvimento de Agentes Inteligentes e de Algoritmos de Negociação, sendo estes termos explicitados no tópico *Glossário*.

Além disso, pretende-se também que seja possível estabelecer a comunicação entre aplicações distribuídas e a utilização via *Web*.

No que diz respeito à comunicação entre aplicações distribuídas, há que referir que os Agentes que representam as diferentes entidades envolvidas na geração de horários mantêm a sua informação privada (ou seja, não partilham o seu conhecimento com os restantes Agentes), constituindo por isso aplicações distribuídas que necessitam de comunicar entre si.

Relativamente à utilização via *Web*, há que referir que permite uma maior portabilidade da aplicação, permitindo que os utilizadores com permissão possam efectuar as operações pretendidas, sem que para tal se tenham de deslocar ao local em que a aplicação está implementada. Este objectivo será realçado quando apresentarmos os casos de uso desta aplicação.

Este sistema que vai ser desenvolvido vai permitir que, por exemplo, a nível da Faculdade de Engenharia, seja possível efectuar a elaboração de horários para as turmas de todos os cursos existentes. No entanto, devido à portabilidade que se pretende impor a este sistema, seria possível utilizá-lo em qualquer instituição de ensino, quer Universitário quer Secundário.

O facto da Arquitectura Multi-Agente ainda estar a ser desenvolvida e de ainda não existirem trabalhos relacionados com a geração semi-automática de horários suportados pela mesma, isso implica um elevado risco, nomeadamente porque não temos garantias de que será possível chegar ao horário óptimo. Além disso, o facto de ser necessário ter o *Router* a correr no mesmo servidor HTTP que as *applets*, implica que a não obtenção de permissões para tal tornará impossível o referido acesso via *Web*.

2. Lista de requisitos do sistema

Antes de apresentarmos os requisitos do sistema, convém referir que a fonte destes requisitos foi o nosso cliente.

1. Geração de horários, tendo em consideração as preferências dos alunos inscritos na turma e as dos seus Professores. Em caso de incompleta impossibilidade, o conflito deverá ser solucionado, privilegiando a turma. Trata-se de um requisito funcional, de prioridade máxima, uma vez que constitui o objectivo principal deste sistema. Quanto ao grau de dificuldade na sua satisfação é elevado, uma vez que é sobre ele que assenta todo o trabalho que irá ser desenvolvido ao longo deste projecto.

2. Contemplar a existência de um Gestor da Informação do Sistema, responsável pela manutenção dos dados relativos aos Professores, alunos, cursos, disciplinas, cadeiras, ramos e planos de estudo, de tal modo que a informação se possa manter sempre actualizada. Além disso, poderá também efectuar consultas a essa mesma informação. O acesso do Gestor da Informação a estas funções deve ser controlado mediante a

introdução do seu *login* e *password*, podendo esta ser alterada pelo mesmo. Este é um requisito funcional desejável, cujo esforço para a sua satisfação está em garantir a consistência dos dados.

3. Permitir que os alunos possam efectuar a sua inscrição numa determinada turma via *Web*, seleccionado as suas preferências para o horário. Além disso, deverá também ser possível consultar a média das preferências da turma até ao momento, assim como os alunos já inscritos na mesma. Este é um requisito funcional necessário, pois é a partir dele que é possível obter a as preferências dos alunos que serão tidas em consideração na geração dos horários. A sua satisfação não requer elevado esforço.

4. Permitir que qualquer *Cibernauta* possa consultar a informação actual de uma determinada turma, assim como o seu horário (que pode ou não já ter sido gerado). Este é um requisito funcional desejável, que não requer elevado esforço para a sua satisfação.

5. Cada Professor deve poder efectuar a manutenção dos seus próprios dados, incluindo-se a consulta dos mesmos. Este é um requisito funcional necessário, uma vez que é a partir das preferências do Professor que também são gerados os horários. O esforço necessário para a sua satisfação está em conseguir obter uma interface que facilite a manutenção dos dados.

6. Permitir que qualquer *Cibernauta* possa consultar o horário de um Professor. Este é um requisito funcional desejável, que não requer elevado esforço para a sua satisfação.

7. Contemplar a existência de um Gestor de Salas, responsável pela manutenção dos dados relativos às salas, o qual também pode efectuar consultas desses mesmos dados. O esforço necessário para a sua satisfação está em conseguir obter uma interface que facilite a manutenção dos dados.

8. Permitir que qualquer *Cibernauta* possa consultar informação relativa às salas. Este é um requisito funcional desejável, que não requer elevado esforço para a sua satisfação.

9. Garantir a persistência dos dados utilizados pelo sistema, armazenando-os em ficheiros. Este é um requisito funcional necessário, cujo esforço para a sua satisfação reside na definição da estrutura dos ficheiros que vão armazenar os dados.

10. Utilizar uma Arquitectura Multi-Agente. Este é um requisito do processo necessário, cujo esforço para a sua satisfação está em prever como é que os agentes se relacionam, assim como efectuar o processamento das mensagens recebidas.

11. Utilizar a plataforma JATLite e a linguagem de programação Java. Este é um requisito do processo necessário, que requer algum esforço na sua satisfação, uma vez que se torna necessário conhecer como se processam as comunicações entre os agentes, assim como a sua ligação ao Router.

12. Implementar mecanismos de segurança do sistema, obrigando à introdução de *login* e *password* correctos para a execução de uma determinada operação. Este é um requisito funcional necessário, que não requer elevado esforço para a sua satisfação.

13. Utilização de *Applets*, de modo a permitir interfaces via *Web*. Este é um requisito funcional necessário, que não requer elevado esforço para a sua satisfação, mas depende de se conseguir executar as *Applets* no mesmo servidor HTTP que o Router.

14. Permitir que o Gestor da Turma altere o horário que acabou de ser gerado, movendo a aula para o novo intervalo de tempo pretendido. Essa alteração só poderá efectivamente ocorrer se existir compatibilidade quer da parte da turma em questão, quer do Professor que está associado à disciplina. Além disso, é também necessário que exista uma sala na qual a aula possa ser leccionada. Este é um requisito funcional necessário, cujo esforço para a sua satisfação é elevado, uma vez que exige um processo semelhante (embora ligeiramente mais simplificado, uma vez que só se trata de saber se é ou não possível efectuar a alteração) ao da geração de horários.

15. Desenvolver um segundo Algoritmo de Negociação para Geração de Horários, que ponha em igualdade de circunstâncias todos os Professores que leccionam aulas do tipo em questão e com uma determinada duração.

Tal como se verá na descrição dos casos de uso, estes requisitos são aí contemplados.

3. Modelo de Casos de Uso

Neste tópico começaremos por apresentar uma visão geral dos casos de uso do sistema, assim como uma descrição dos seus actores. Em seguida, apresentamos uma descrição detalhada dos casos de uso, dividindo-os em casos de uso exteriores e interiores ao sistema.

Os casos de uso exteriores ao sistema são aqueles em que os actores não são agentes, enquanto que os casos de uso interiores ao sistema são aqueles em que os seus actores são interiores ao sistema, correspondendo neste caso a agentes.

Convém ainda referir que a razão pela qual os casos de uso estão bastante divididos se deve ao facto do sistema ser Multi-Agente e, como tal, os casos de uso estarem distribuídos por esses mesmos agentes.

3.1. Visão Geral do Sistema

Tal como se pode verificar, o Gestor da Informação do sistema é responsável pela gestão dos dados gerais do sistema (que, tal como se verá mais à frente, são os dados relacionados com os Professores, alunos, cursos, disciplinas, cadeiras, ramos e planos de estudo).

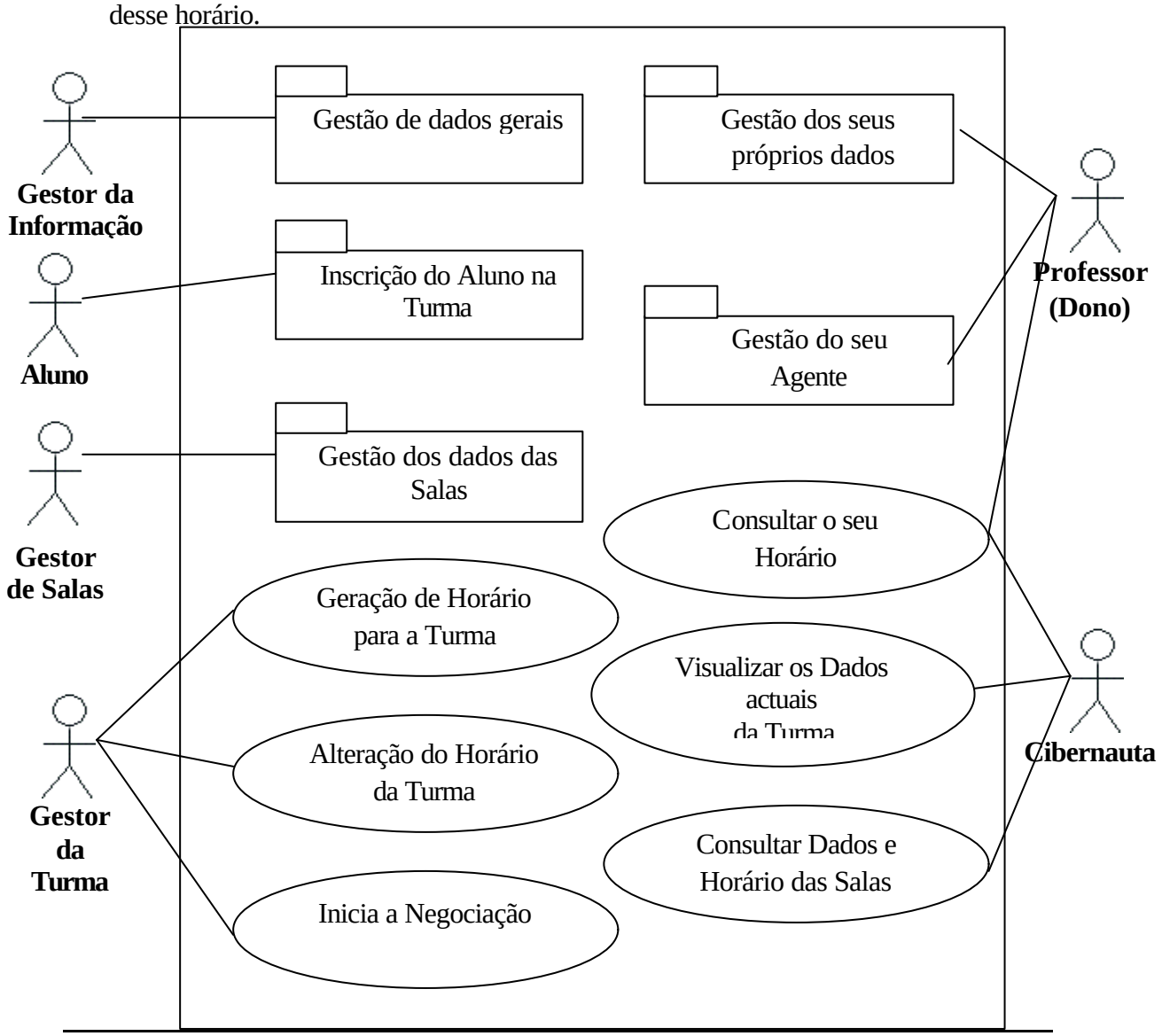
Relativamente ao Aluno, pode efectuar a sua inscrição numa determinada turma, tendo para tal que seleccionar a turma em que se pretende inscrever, assim como as suas preferências para esse horário.

O Gestor de Salas é o responsável pela gestão dos dados das salas.

O Professor é responsável por efectuar a gestão dos seus próprios dados, assim como a gestão do seu agente. Além disso, o Professor também pode consultar o seu horário.

Relativamente ao Cibernauta, pode também consultar o horário de um determinado Professor, visualizar os dados actuais de uma determinada turma (a média das preferências dos alunos inscritos, assim como os alunos já inscritos), assim como consultar dados relativos às salas.

O Gestor da Turma é responsável por seleccionar a turma para a qual pretende gerar o horário, sendo ele que dá ordem para que essa geração se inicie. É também possível que o Gestor da Turma efectue alterações no horário gerado, as quais podem ou não ser aceites, dependendo se melhoram ou não a utilidade desse horário.



3.2. Actores do Sistema

Este sistema contempla vários actores, os quais serão descritos seguidamente, por ordem alfabética.

3.2.1 Aluno

Um Aluno é uma pessoa que frequenta um determinado curso no estabelecimento de ensino no qual o sistema está implementado.

3.2.2 Applet Horário Professor

A Applet Horário Professor é um agente que permite visualizar o horário de um determinado Professor, via *Web*.

3.2.3 Applet Inscrição

A Applet Inscrição é um agente que permite que um aluno possa efectuar a sua inscrição numa determinada turma, seleccionado para tal as suas preferências.

3.2.4 Applet Professor

A Applet Professor é um agente que permite que um Professor possa efectuar a gestão dos seus próprios dados e do seu agente, via *Web*.

3.2.5 Applet Salas

A Applet Salas é um agente que permite que qualquer Cibernauta possa consultar informação acerca das salas.

3.2.6 Applet Turma

A Applet Turma é um agente que permite que qualquer Cibernauta possa consultar os dados actuais (média das preferências e alunos inscritos) relativos a uma determinada turma.

3.2.7 Cibernauta

O Cibernauta é uma pessoa que acede ao sistema via *Web*.

3.2.8 Gestor da Informação

O Gestor da Informação é o responsável pela gestão dos dados gerais do sistema, ou seja, os que estão relacionados com aos Professores, alunos, cursos, disciplinas, cadeiras, ramos e planos de estudo.

3.2.9 Gestor da Turma

O Gestor da Turma é o responsável por iniciar a negociação para a geração do horário da turma em questão. Além disso, é também ele que pode efectuar alterações no horário obtido, as quais poderão ou não ser aceites, dependendo da disponibilidade da turma e do Professor que lecciona a disciplina em questão, assim como da existência ou não de sala onde a aula pode ter lugar.

3.2.10 Gestor de Salas

O Gestor de Salas é o responsável pela gestão dos dados das salas.

3.2.11 Outros Agentes

Os Outros Agentes representam agentes que, em determinada circunstância, necessitam de dispor da mesma funcionalidade que outros actores. Em cada uma das circunstâncias em que aparecem, serão especificados os actores que representam.

3.2.12 Professor

O Professor é uma pessoa que lecciona determinadas disciplinas no estabelecimento de ensino no qual o sistema está a ser utilizado.

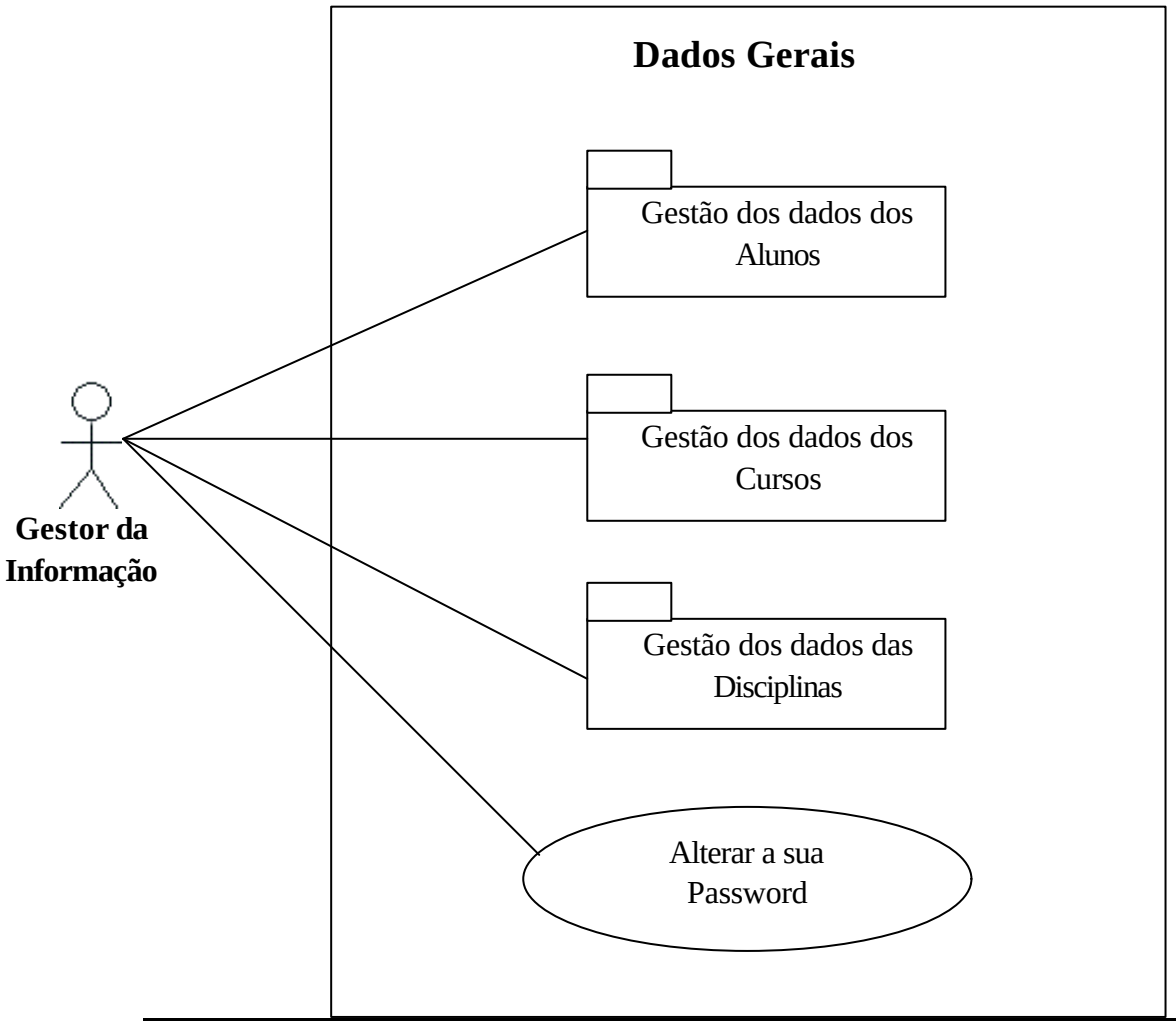
3.2.13 Turma que inicia a negociação

A Turma que inicia a negociação é uma turma que está a desempenhar o papel de chefia de negociação entre os vários intervenientes. Além disso, também guarda aspectos pedagógicos que devem ser respeitados na geração do horário de uma turma.

3.3. Casos de Uso Exteriores ao Sistema

Seguidamente apresentamos os casos de uso exteriores ao sistema, acompanhados de uma breve descrição, assim como os actores que os desempenham.

3.3.1 Casos de uso associados aos dados gerais do sistema



Tal como se pode verificar, o Gestor da Informação da aplicação tem a seu cargo efectuar a gestão dos dados dos alunos, dos cursos e das disciplinas. Para tal, torna-se necessário que antes de poder efectuar qualquer uma destas operações, introduza o *login* e a *password* adequados.

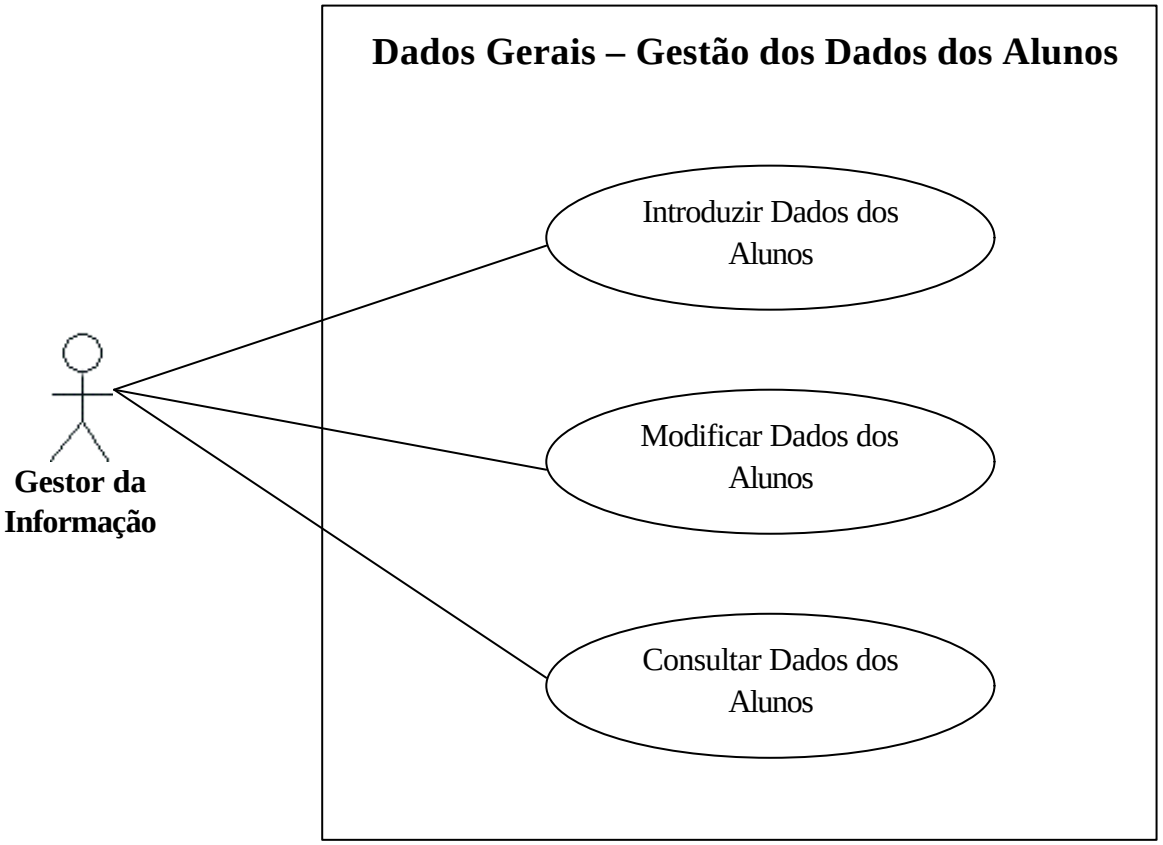
Além disso, pode também alterar a sua *password*.

Convém referir que no Diagrama acima apresentado não foram incluídos os casos de uso para este actor relativos aos Professores, Cadeiras, Ramos e Planos de Estudo, uma vez que o Diagrama se tornaria demasiado grande e não haveria essa necessidade, visto serem idênticos. Por este motivo, apenas foram apresentados três tipos de dados, sendo os restantes análogos a estes.

Estes casos de uso correspondem ao requisito número 2 da lista de requisitos anteriormente referida.

3.3.1.1 Gestão dos dados dos alunos

Este ponto indica quais os casos de uso associados ao pacote *Gestão dos dados dos Alunos* do diagrama de casos de uso anterior.

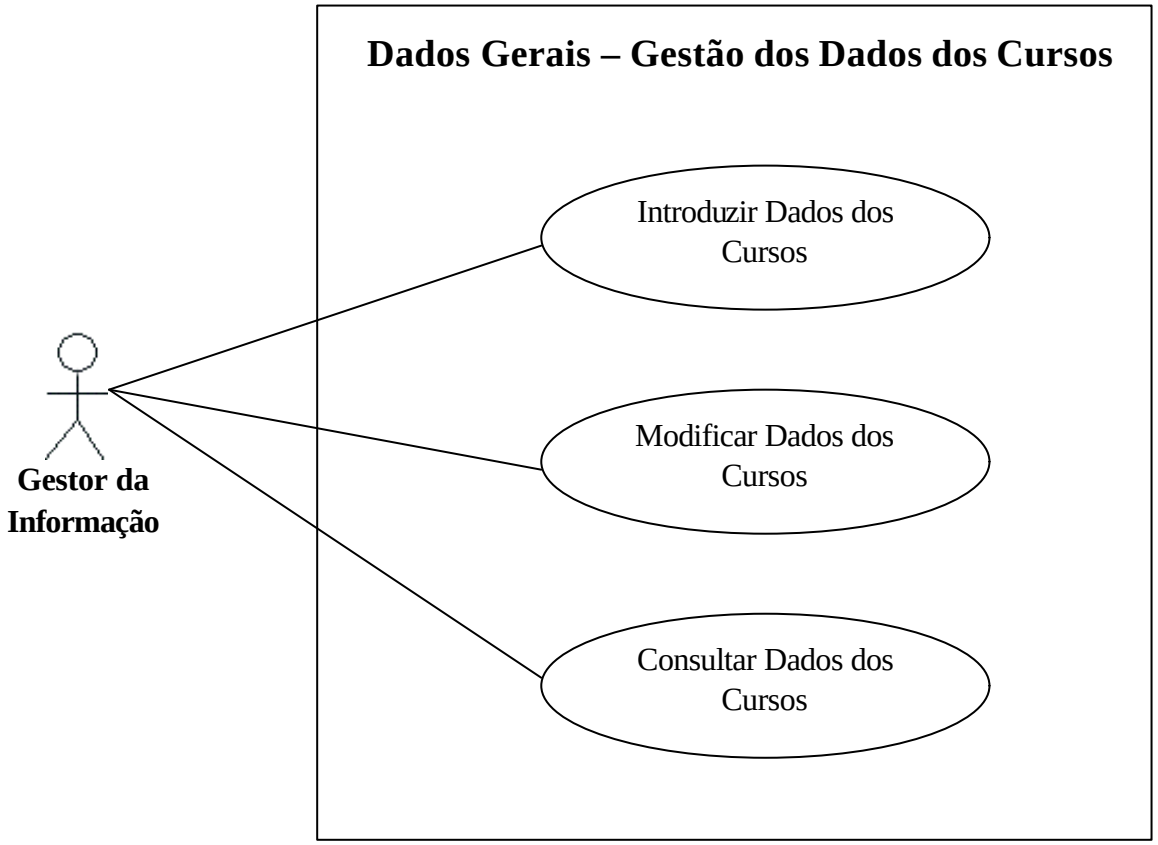


Tal como se pode verificar, o Gestor da Informação desta aplicação no que diz respeito à *Gestão dos Dados dos Alunos*, pode introduzir, modificar e consultar dados relativos a alunos.

3.3.1.2 Gestão dos dados dos cursos

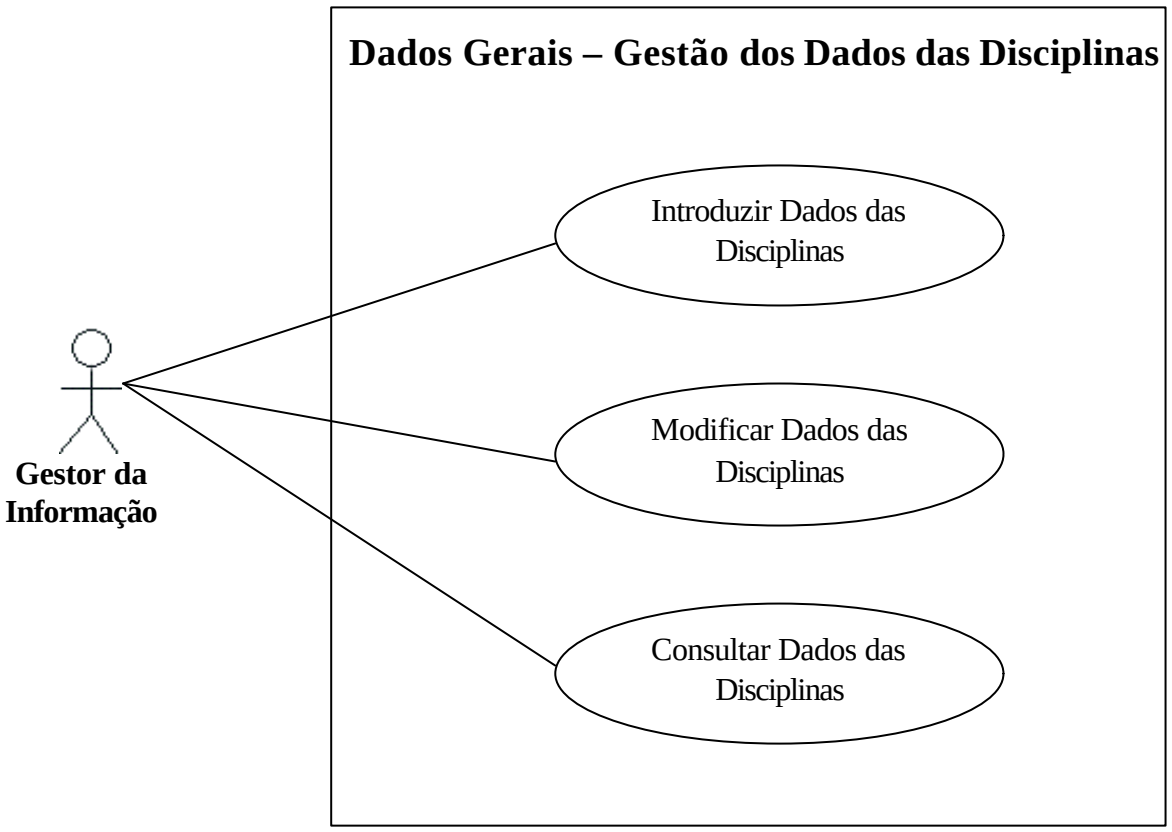
Este ponto indica quais os casos de uso associados ao pacote *Gestão dos dados dos Cursos* do diagrama de casos de uso apresentado em **3.3.1**.

Tal como se pode verificar, o Gestor da Informação desta aplicação no que diz respeito à *Gestão dos dados dos Cursos* pode executar operações semelhantes às efectuadas para a *Gestão dos dados dos Alunos*: introdução, alteração e consulta dos dados relativos aos cursos.



3.3.1.3 Gestão dos dados das disciplinas

Este ponto indica quais os casos de uso associados ao pacote *Gestão dos dados das Disciplinas* do diagrama de casos de uso apresentado em **3.3.1**.

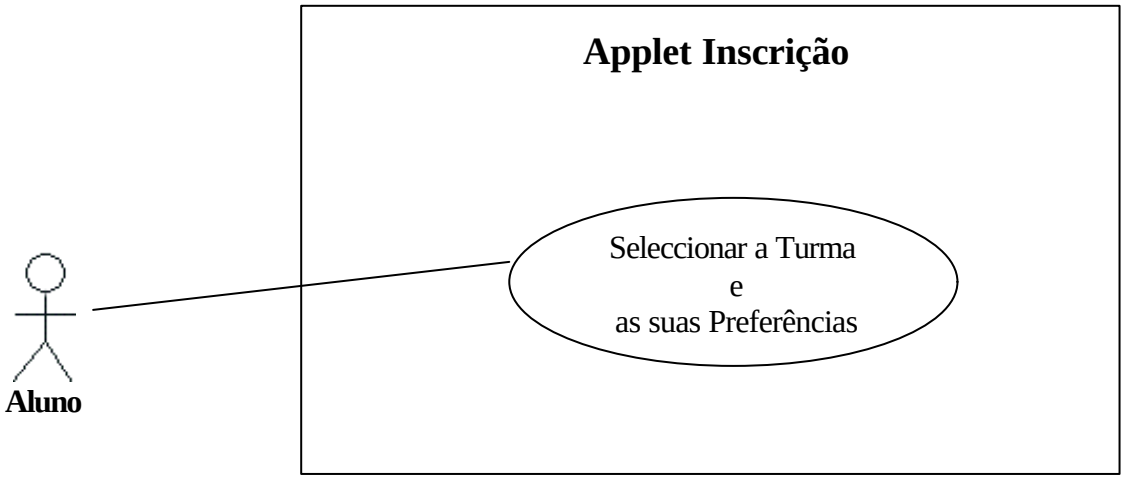


Tal como se pode verificar, o Gestor da Informação desta aplicação no que diz respeito à *Gestão dos dados das Disciplinas* pode executar operações semelhantes às efectuadas para a *Gestão dos dados dos Alunos*: introdução, alteração e consulta dos dados das disciplinas.

3.3.2 Caso de uso da Applet Inscrição

Tal como se pode verificar, é através de uma Applet de Inscrição que o Aluno (via *Web*) vai poder seleccionar a turma em que se pretende inscrever, assim como as suas preferências para esse horário. Para tal, é necessário que antes de mais seja validado o seu *login* assim como a sua *password*.

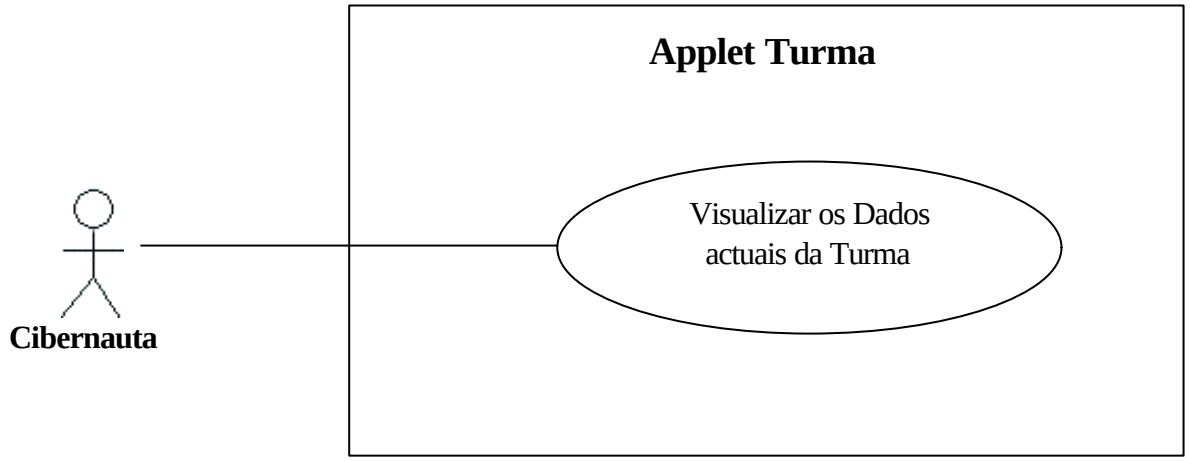
Este caso de uso corresponde ao requisito número 3 da lista de requisitos anteriormente referida.



3.3.3 Caso de uso da Applet Turma

Tal como se pode verificar, é através da Applet Turma que é possível que o Cibernauta possa visualizar os dados actuais da turma, nomeadamente a média das preferências até ao momento, assim como os alunos que já fizeram a sua inscrição.

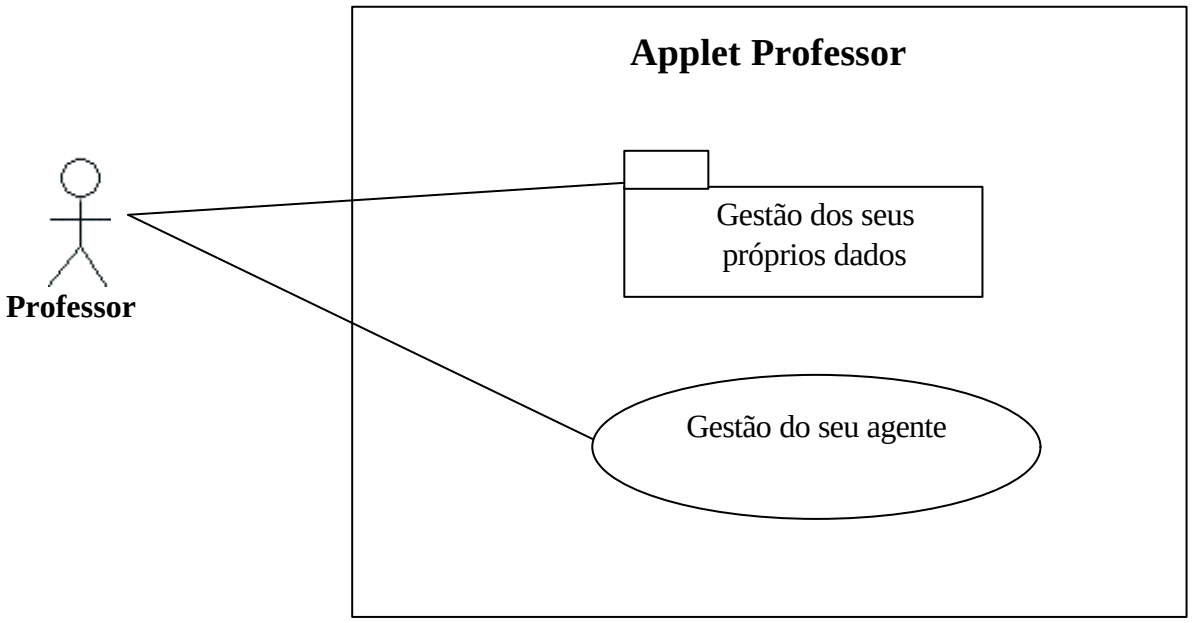
Este caso de uso corresponde ao requisito número 4 da lista de requisitos anteriormente referida.



3.3.4 Casos de uso da Applet Professor

Tal como se pode verificar, é através da Applet Professor que é possível ao Professor efectuar a gestão dos seus próprios dados, assim como gerir o seu agente. Para tal é necessário que antes seja validado o *login* e a *password* do Professor em questão.

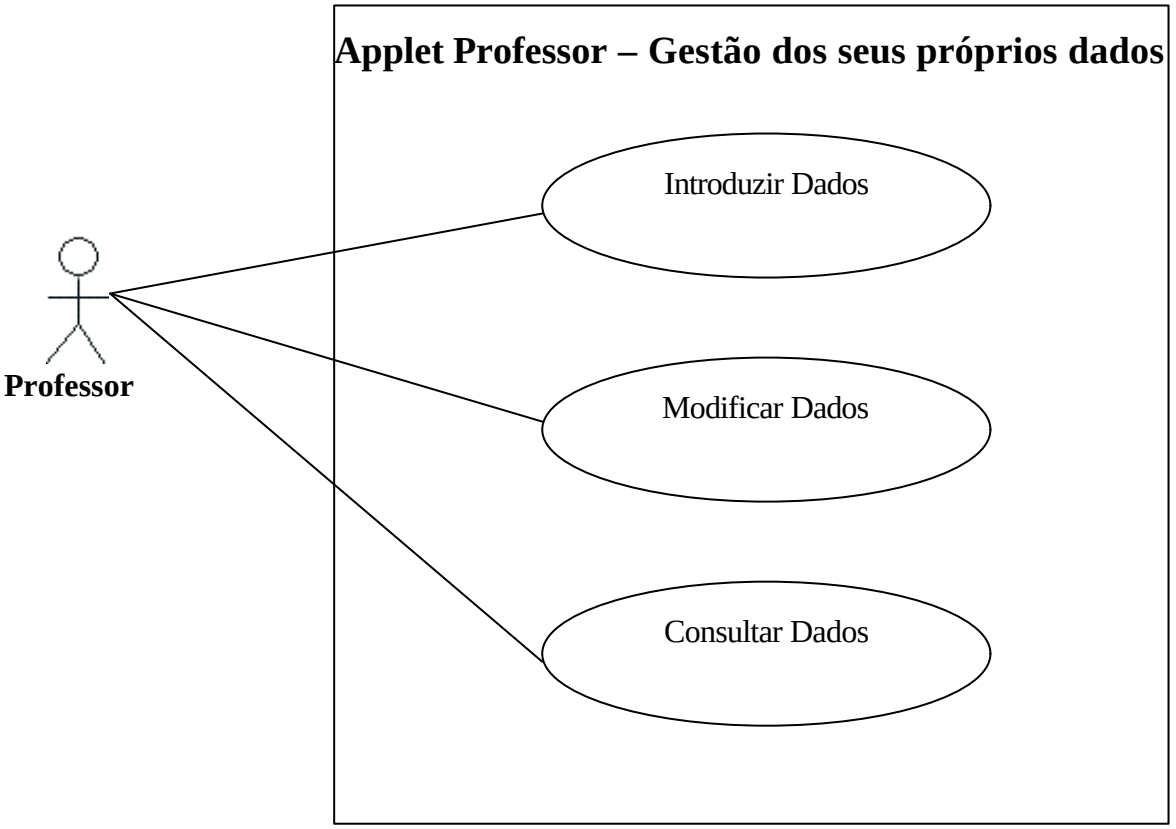
Este caso de uso corresponde ao requisito número 5 da lista de requisitos anteriormente referida.



3.3.4.1 Gestão dos seus próprios dados

Este ponto indica quais os casos de uso associados ao pacote *Gestão dos seus próprios dados* do diagrama de casos de uso apresentado em **3.3.4**.

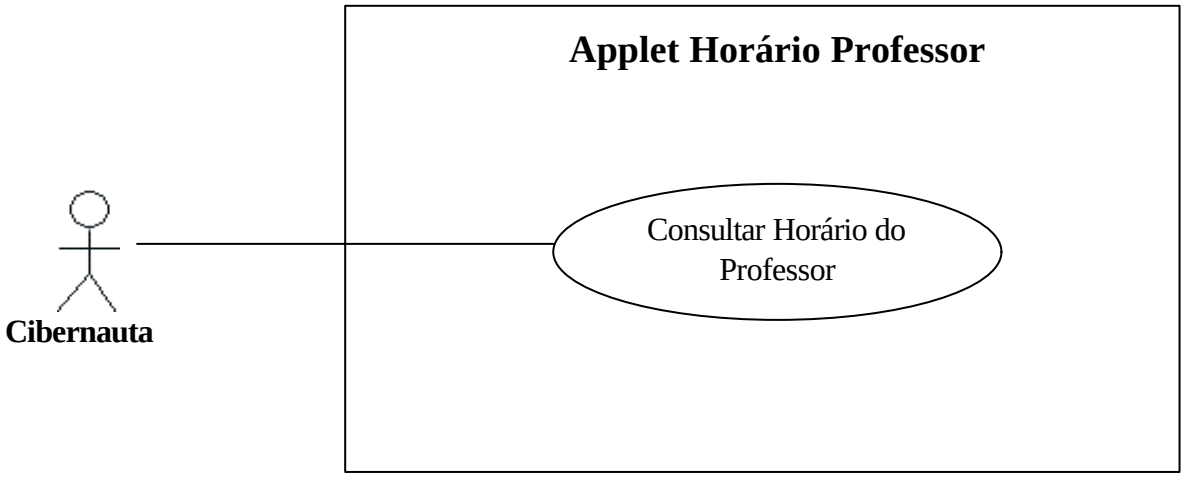
Tal como se pode verificar, o Professor no que diz respeito à *Gestão dos seus próprios dados* pode introduzir, modificar e consultar os seus próprios dados.



3.3.5 Caso de uso da Applet Horário Professor

Tal como se pode verificar, a Applet Horário Professor permite ao Cibernauta consultar o horário do Professor em questão.

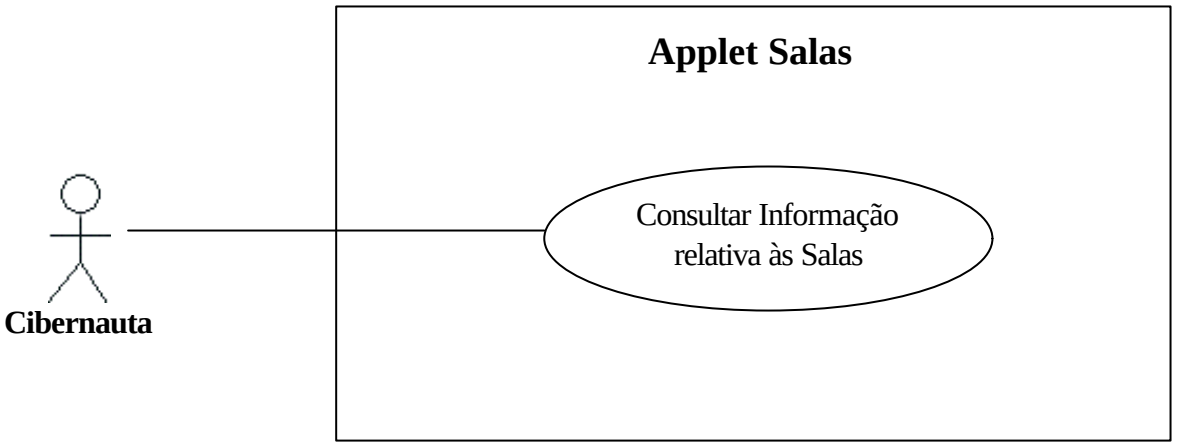
Este caso de uso corresponde ao requisito número 6 da lista de requisitos anteriormente referida.



3.3.6 Caso de uso da Applet Salas

Tal como se pode verificar, a Applet Salas permite ao Cibernauta consultar informação relativa a salas.

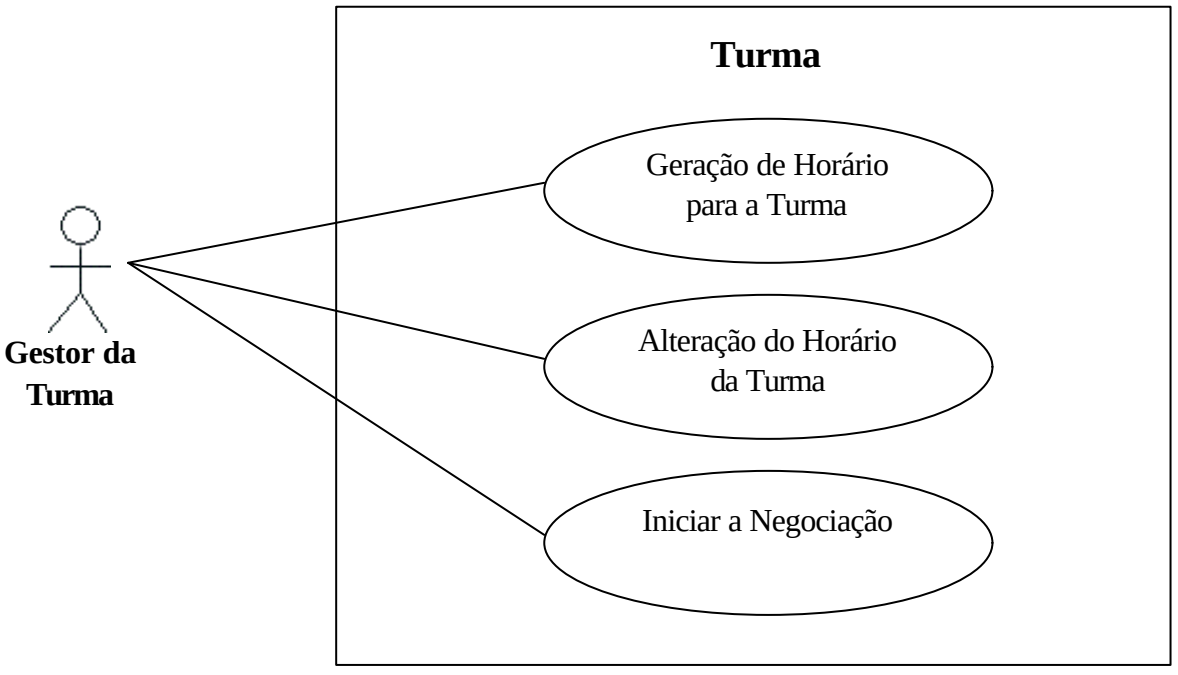
Este caso de uso corresponde ao requisito número 8 da lista de requisitos anteriormente referida.



3.3.7 Casos de uso da Turma

Tal como se pode verificar, o Gestor da Turma é quem faz a geração do horário para a turma em questão, sendo também ele quem pode efectuar alterações no horário obtido, as quais podem ser ou não aceites (dependendo da disponibilidade da turma, do Professor e da existência ou não de sala). Além disso, é também o Gestor da Turma quem inicia a negociação para a geração do horário para a turma em questão.

Este caso de uso corresponde ao requisito número 14 da lista de requisitos anteriormente referida.

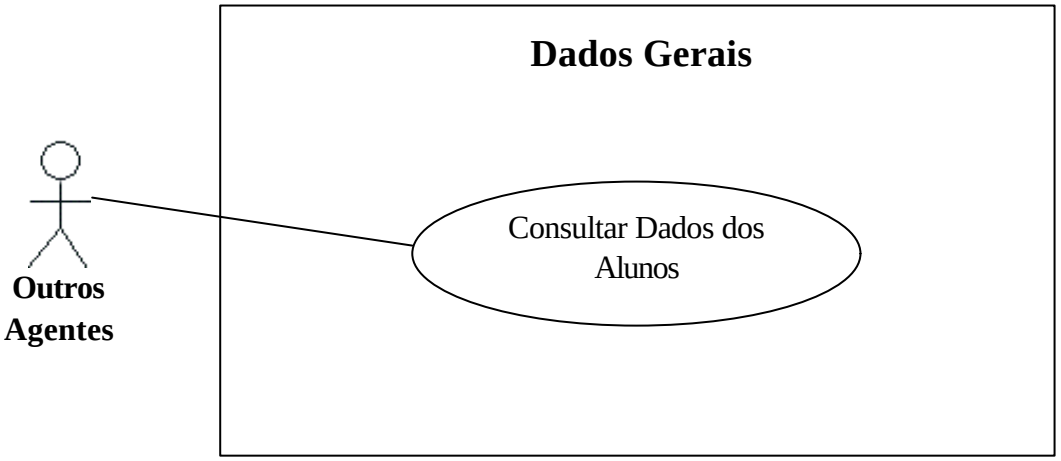


3.4. Casos de Uso Interiores ao Sistema

Seguidamente apresentamos os casos de uso interiores ao sistema, acompanhados de uma breve descrição, assim como os actores que os desempenham.

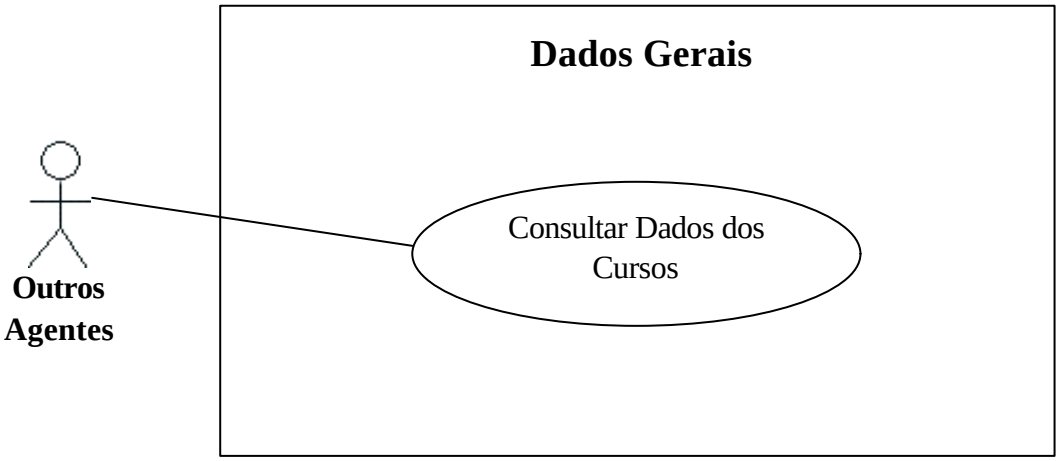
3.4.1 Caso de Uso da Gestão dos Dados dos Alunos

Tal como se pode verificar, a consulta de dados dos alunos pode ser efectuada por Outros Agentes, que neste caso podem ser os Professores, a Turma e sempre que em qualquer *Applet* se tornar necessário a introdução de *login* e *password* do aluno.



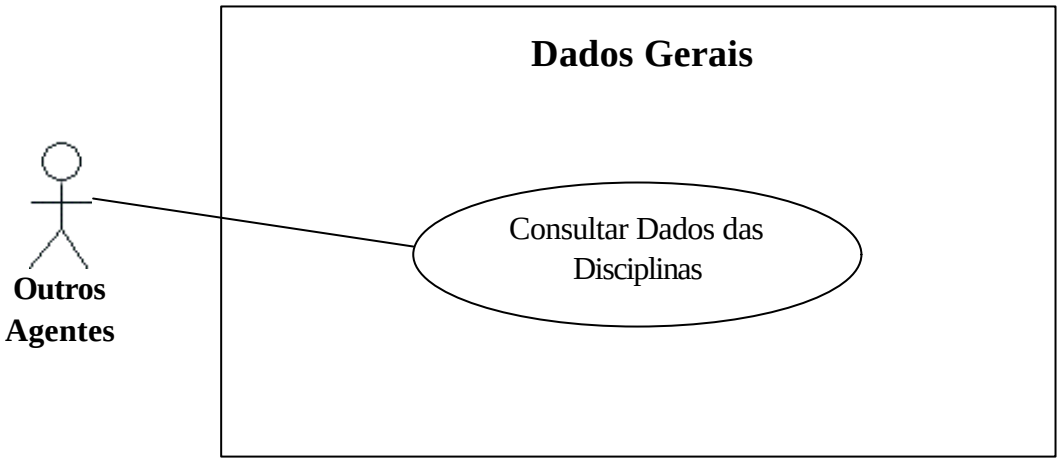
3.4.2 Caso de Uso da Gestão dos Dados dos Cursos

Tal como se pode verificar, a consulta de dados dos cursos pode ser efectuada por Outros Agentes, que neste caso podem ser Professores (para saberem quais as disciplinas que leccionam) e a Turma (para saber quais as disciplinas que tem).



3.4.3 Caso de Uso da Gestão dos Dados das Disciplinas

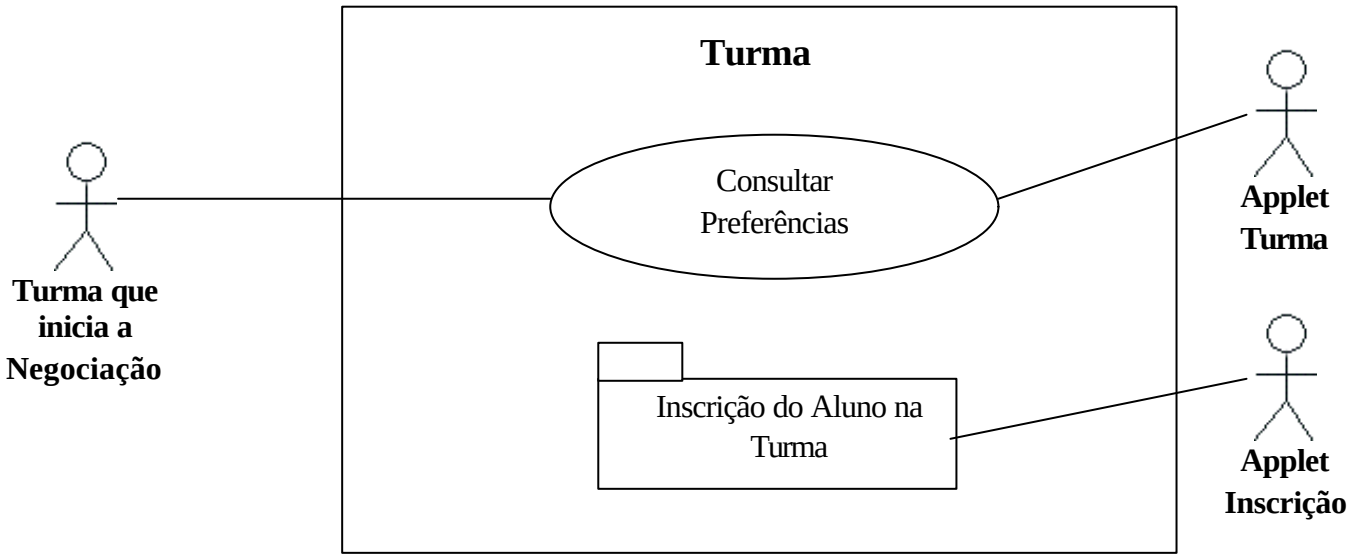
Tal como se pode verificar, a consulta de dados das disciplinas pode ser efectuada por Outros Agentes, que neste caso podem ser Professores (para saberem informação acerca das disciplinas que leccionam) e a Turma (para obter informação acerca das disciplinas que lecciona).



3.4.4 Casos de uso associados à Turma

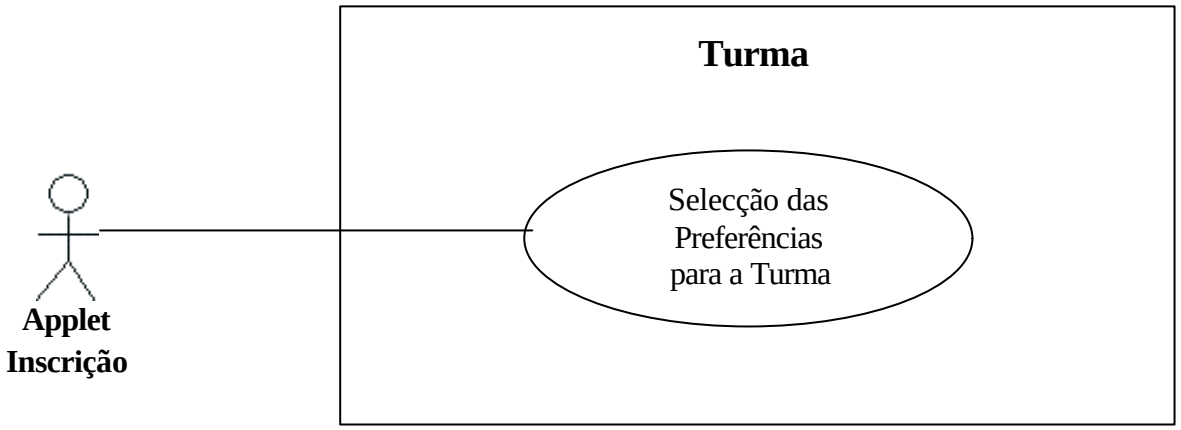
Tal como se pode verificar, a Turma que inicia a Negociação, assim como a Applet Turma podem consultar a média das preferências da turma.

Relativamente à Applet Inscrição, permite que o aluno se inscreva na turma pretendida, seleccionando as suas preferências.



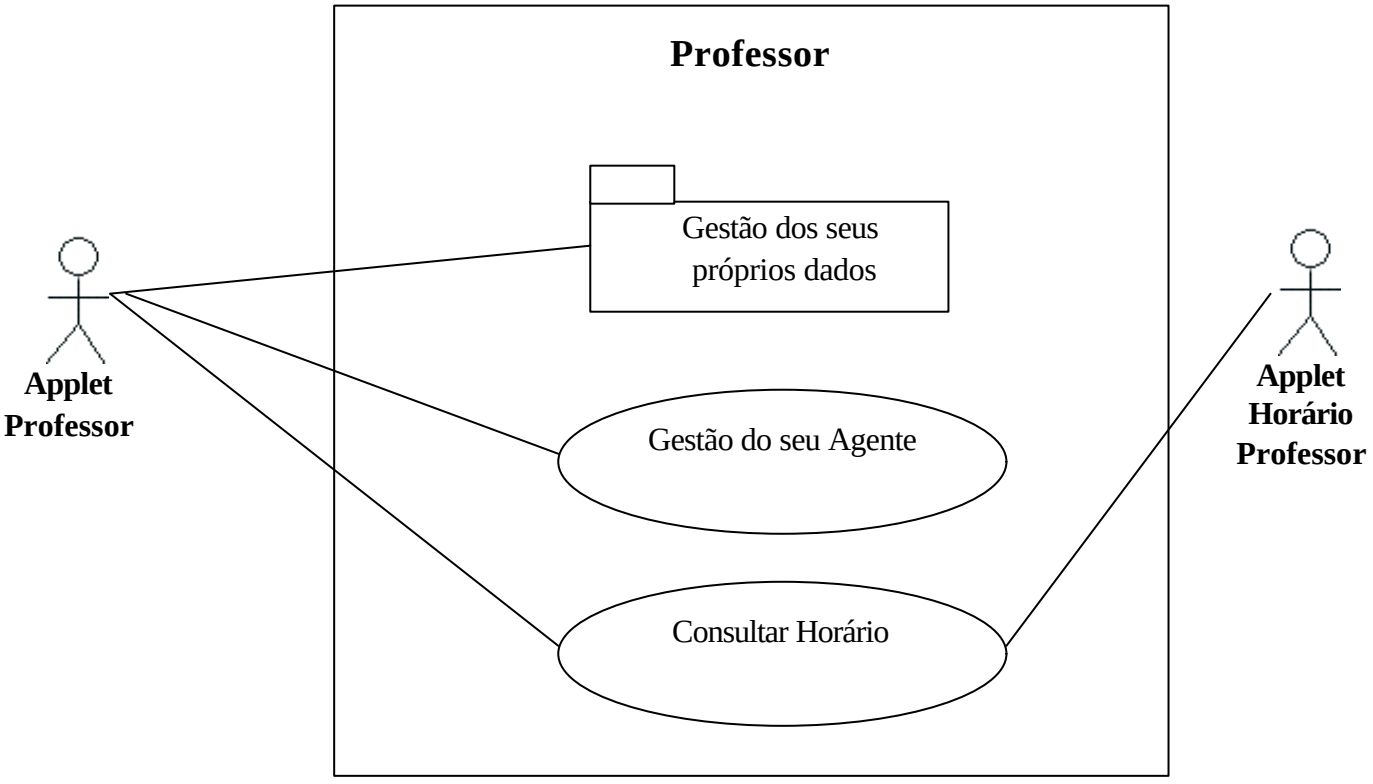
3.4.4.1 Inscrição do Aluno na Turma

Este ponto indica quais os casos de uso associados ao pacote *Inscrição do Aluno na Turma* do diagrama de casos de uso apresentado em **3.4.4**.



Tal como se pode verificar, a Applet Inscrição efectua a selecção das preferências do aluno para o horário da turma em questão, concretizando a acção de inscrição iniciada pelo aluno.

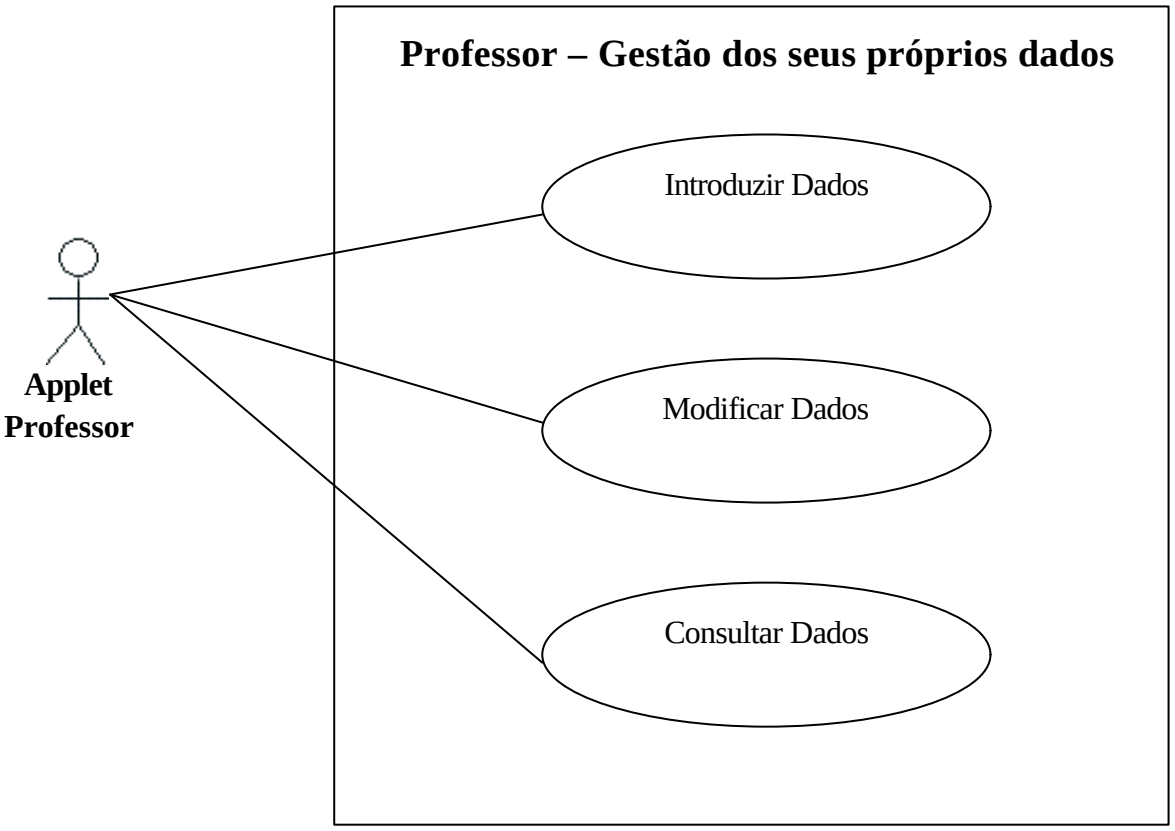
3.4.5 Casos de uso associados ao Professor



Tal como se pode verificar, existe um actor designado por Applet Professor que pode efectuar a gestão dos seus próprios dados, assim como a gestão do seu agente. Além disso também pode consultar o seu horário, sendo que esta última funcionalidade está também disponível para a Applet Horário Professor.

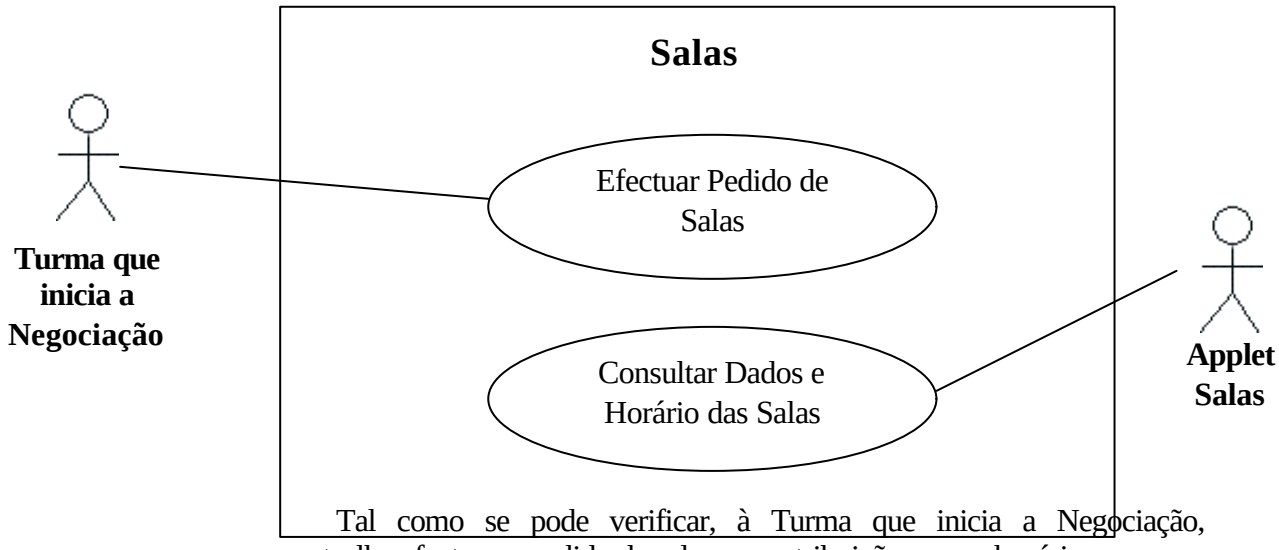
3.4.5.1 Gestão dos seus próprios dados

Este ponto indica quais os casos de uso associados ao pacote *Gestão dos seus próprios dados* do diagrama de casos de uso apresentado em **3.4.5**.



Tal como se pode verificar, a Applet Professor no que diz respeito à *Gestão dos seus próprios dados* pode executar operações de introdução, modificação e consulta dos mesmos.

3.4.6 Casos de uso associados às Salas



Tal como se pode verificar, à Turma que inicia a Negociação, compete-lhe efectuar o pedido de salas para atribuição ao seu horário.

Quanto à Applet Salas pode efectuar a consulta dos dados relativos às salas.

3.5 Exemplos de Interfaces

Relativamente à Versão 1.0 convém referir que os exemplos de interfaces que aqui eram apresentados foram retirados, uma vez que grande parte dessas interfaces foram substituídas por outras (devido a opções tomadas durante o desenvolvimento do Projecto).

Sendo assim, remete-se este ponto para os vários Manuais do utilizador desenvolvidos para esta Aplicação.

4 Requisitos Suplementares

Além dos requisitos do sistema que já foram anteriormente referidos, convém também indicar os requisitos não funcionais que também devem ser contemplados por este sistema:

1. *Usabilidade*

   No que diz respeito a este requisito, as interfaces devem ter em consideração o tipo de utilizador a que se destinam, proporcionando-lhe facilidade de aprendizagem e de utilização, o que implica que ao longo do desenvolvimento do sistema se dê importância ao modo como estão a ser desenvolvidas as interfaces.

   Além disso, é também necessário ter em consideração que, no final, o sistema deverá ser acompanhado de um manual do utilizador, que deverá facilitar a utilização do sistema àqueles a quem se destina.

   Para facilitar a utilização do sistema àqueles a quem se destina, será efectuada uma demonstração na altura da entrega do sistema.

2. *Fiabilidade*

   No que diz respeito a este requisito ficou acordado quais os tipos de erro que devem ser detectados, devendo estes ser reportados ao utilizador de forma a que este seja capaz de perceber o que sucedeu. Sendo assim, o utilizador sente-se mais confiante ao utilizar o sistema, uma vez que sabe que se cometer algum erro, o sistema lho indicará, apontando-lhe para a razão pela qual o erro sucedeu.

3. *Desempenho*

   Relativamente a este sistema a única restrição que se coloca a nível do desempenho é a obtenção dos melhores horários, que maximizam as preferências dos alunos inscritos e dos Professores que lhe estão associados.

   No que diz respeito ao tempo de resposta, à utilização de memória ou à velocidade das transacções, trata-se de aspectos que são irrelevantes para este sistema, e que não intervêm na avaliação do seu desempenho. No entanto, pretende-se que não exista um gasto desnecessário nem excessivo dos recursos disponíveis.

4. *Supportability*

Relativamente a este ponto ficou acordado que o sistema seria entregue juntamente com alguns dados (de Professores, alunos, cursos, disciplinas, cadeiras, ramos, planos de estudo e salas), de tal modo que pudessem ser testadas todas as suas funcionalidades.

Relativamente à manutenção do sistema, ficou acordado que esta seria efectuada sempre que necessário, o que só deverá suceder caso ocorram alterações na Lei que tragam restrições relativas à carga horária diária ou relacionadas com aspectos pedagógicos.

5 Modelo de objectos do domínio

Neste ponto apresentamos uma descrição do diagrama de classes do nosso sistema, o qual se encontra em *Anexo.*

Uma Pessoa é caracterizada pelo nome e pelo *email*. Tal como se pode verificar, este sistema contempla dois tipos de Pessoas: o Aluno e o Professor.

O Aluno é caracterizado por um número – *numero* - , pelo seu *login* e *password* assim como pelo ano do curso - *anoCurso* – que frequenta.

O Professor é caracterizado pela sua *sigla*, pelo *departamento* a que pertence e pela sua *categoria*. Além disso, tem também associado as suas preferências.

Relativamente a um aluno, este pode estar inscrito num dado instante apenas numa turma, podendo esta ter vários alunos inscritos. A Turma guarda a sua *sigla*, o ano a que se refere, assim como a média das preferências dos alunos nela inscritos. Além disso, a Turma dispõe de um método que permite determinar o número de alunos que nela estão inscritos – *numeroAlunos()*.

Relativamente a uma Sala é guardado o seu nome – *sala* – o seu *tipo*, a sua *capacidade*, a *cor do quadro*, assim como o seu *telefone*. Além disso, é também guardado o tempo que demora a deslocação de uma determinada sala a todas as restantes.

Quer a Turma, quer o Professor quer a Sala têm várias Ocupações, sendo que uma determinada ocupação apenas diz respeito a uma Turma, a um Professor e a uma Sala. Quanto a uma Ocupação é guardado o dia da semana em que ocorre – *diaSemana* - a sua hora de início – *horaInicio* - a sua duração – *duracao* – assim como a sua descrição – *descricao*.

Um Aluno está inscrito num Curso, sendo que um Curso tem vários Alunos inscritos.

Relativamente a um Curso e a um Ramo é guardada a sua *sigla*, assim como a sua designação- *designacao*.

Um Curso é constituído por vários Ramos, sendo que um Ramo apenas faz parte de um Curso. Entre um Curso e um Ramo poder-se-á referir que a relação existente é a de que um Ramo diz respeito a vários anos do Curso. Além disso, um Curso tem várias Turmas, sendo que uma Turma pertence a um único Curso.

Um ano de um determinado Ramo fica caracterizado por um Plano de Estudos.

Um Plano tem associadas várias Disciplinas, podendo uma Disciplina estar associada a vários Planos.

Uma Disciplina é caracterizada pela sua *sigla*, *designacao*, pelo número de aulas teóricas que vai ter – *nTeoricas* - assim como pela sua duração – *duracaoTeorica*. Além disso, é também caracterizada pelo número de alas práticas que vai ter – *nPraticas* – assim como pela sua duração – *duracaoPratica*. Da relação

existente entre um Plano e uma Disciplina nasce um atributo que permite saber se se trata ou não de uma disciplina opcional.

Um Professor pode leccionar várias Disciplinas teóricas, assim como várias Disciplinas práticas, sendo que uma Disciplina apenas poderá ter um Professor que leccione as aulas teóricas, assim como um único Professor que leccione as aulas práticas. No entanto, é possível que um Professor leccione quer as aulas teóricas quer as aulas práticas de uma determinada Disciplina.

É de notar que uma vez que todas as classes existentes neste diagrama utilizam funções de manutenção dos seus próprios dados, estas não foram incluídas de modo a não tornar a sua leitura tão complicada. Estas operações de manutenção dos dados correspondem a operações de inserção, alteração e consulta de dados.

6 Algoritmo para a Geração de Horários 1

Seguidamente, vamos descrever o primeiro Algoritmo que irá ser utilizado para a geração de horários, o qual foi acordado com o nosso cliente. O fluxograma que descreve este Algoritmo encontra-se em *Anexo*.

Inicialmente, a Turma que inicia a negociação começa por pedir uma possibilidade às turmas e aos Professores, avaliando-se em seguida o valor de cada uma das propostas recebidas, através de uma função de utilidade, cujos parâmetros de avaliação podem ser definidos pelo utilizador. Se houve melhoria da solução, repete-se o procedimento anteriormente descrito. Caso contrário, a solução válida é a obtida na iteração anterior.

Tendo-se aceitado uma solução, verifica-se se ainda existem mais aulas teóricas para atribuir ao horário da turma em questão: se sim, repete-se todo o processo descrito até aqui; caso contrário, verifica-se se o horário é ou não aceite pelo Gestor da Turma. Se o horário não for aceite pelo Gestor da Turma, é escolhida a hipótese com menor valor no horário, reiniciando-se o processo descrito, desde o pedido de possibilidades. Se o horário for aceite, a negociação apenas se passa a efectuar entre uma turma e os seus Professores, uma vez que neste momento apenas falta colocar no horário as aulas práticas.

Relativamente à colocação das aulas práticas no horário de uma turma, os passos seguidos são análogos ao que é efectuado para as aulas teóricas, diferindo apenas no caso do horário obtido não ser aceite pelo Gestor da Turma, em que podem ocorrer duas situações: escolher a hipótese com menor valor relativamente às aulas práticas ou permitir que o Gestor da Turma efectue alterações no horário, bastando para tal mover a aula que pretende alterar para a hora pretendida, podendo essa alteração ser ou não sucedida, o que depende da disponibilidade quer do Professor quer da existência de salas onde a aula pode ter lugar. Quando o horário for definitivamente aceite, o Algoritmo é dado por terminado.

7 Algoritmo para a Geração de Horários 2

Seguidamente, vamos descrever o segundo Algoritmo que irá ser utilizado para a geração de horários, o qual foi acordado com o nosso cliente.

O objectivo deste segundo Algoritmo é colocar em igualdade todos os Professores que leccionam aulas de um determinado tipo para a Turma Negociadora, com a mesma duração.

Sendo assim, são pedidas disponibilidades a todos os intervenientes, as quais são depois confrontadas.

As propostas são depois ordenadas, ficando-se com a melhor. Em caso de empate, opta-se pela proposta do Professor que tem menor valor actual para o seu horário, de modo a não o fazer afastar demasiado das suas preferências.

8 Integração com sistemas existentes

Uma vez que não se pretende que este sistema seja integrado noutros já existentes, não é necessário estabelecer a comunicação com os mesmos, a fim de se obter a informação necessária.

Sendo assim, este sistema pode existir de forma independente e autónoma, estando apenas limitado aos seus próprios dados e definições.

9 Validações junto do Cliente

Foi necessário efectuar alguns esclarecimentos junto do cliente, os quais passamos a apresentar:

1. *Manter as preferências privadas*

   Uma das questões que colocámos está relacionada com o facto das preferências dos agentes serem privadas ou, antes pelo contrário, cada agente dar a conhecer aos restantes as suas preferências.

   No entanto, a razão dada para se optar pela primeira possibilidade é muito forte para se pensar em optar pela segunda: ao manter as suas preferências privadas, o agente apenas vai dando a conhecer as suas preferências à medida que lhe são pedidas disponibilidades. Sendo assim, o agente pode conseguir obter uma solução melhor para si, o que poderia não suceder se os restantes agentes conhecessem as suas preferências, uma vez que se poderia prejudicar uma das partes para beneficiar o todo.

2. *Necessidade de visualização das interacções*

   Relativamente a este ponto coloca-se a possibilidade de se permitir a visualização das interacções da negociação ou não.

   Este aspecto foi deixado ao nosso critério para a fase de desenvolvimento, sendo de notar que a sua inclusão seria uma valia, uma vez que permitira ao utilizador ter a percepção da evolução da negociação entre os agentes, assim como perceber qual foi o caminho seguido para se chegar à solução final.

3. *Interface Web para a Base de Dados Geral*

   Relativamente a este ponto coloca-se a possibilidade da interface para a gestão da base de dados geral ser ou não via *Web*.

   No entanto, uma vez que existe apenas uma pessoa responsável pela gestão dos dados da base de dados geral, não se torna necessário disponibilizar um acesso via *Web*.

4. *Situações de completa incompatibilidade*

   Um outro problema que colocámos está relacionado com o modo como deveríamos de solucionar situações de incompleta incompatibilidade.

   De acordo com a opinião do nosso cliente, uma vez que os Professores ligados ao estabelecimento de ensino em questão têm por obrigação dar aulas, é necessário que assegurem a realização das aulas que lhes estão atribuídas. Sendo

assim, em caso de completa impossibilidade, o conflito deve ser resolvido em favor da turma, uma vez que a esta têm que ser asseguradas a leccionação das aulas.

10 Glossário

Neste tópico apresentamos os termos do vocabulário utilizado neste sistema por ordem alfabética, assim como uma breve descrição do seu significado, de tal modo que em posteriores referências aos mesmos se compreenda o seu significado.

à Agentes

Os Agentes são entidades computacionais que permitem efectuar tarefas custosas (quer em tempo quer em "poder"), libertando assim os humanos desse tipo de tarefas representando-o nas mesmas.

Estas entidades são autónomas (ou seja, mantêm o controlo sobre as suas acções e estados internos), interagem com outros agentes, agem de modo a alcançarem os "seus" objectivos, comunicando apenas factos verdadeiros aos outros agentes.

à Algoritmos de Negociação

Os Algoritmos de Negociação constituem procedimentos a seguir para que seja possível obter um comportamento conjunto dos agentes desejado.

Por um lado, servem para que seja possível atingir objectivos comuns (no nosso caso, a obtenção de horários quer para as turmas quer para os Professores); por outro lado servem para que os agentes satisfaçam os seus próprios objectivos (no nosso caso, a maximização das preferências das turmas e dos Professores).

à ANS

ANS é a abreviatura de *Agent Name Service* e que é um serviço que permite obter o endereço IP de qualquer máquina, a partir do seu nome.

à Cibernauta

O Cibernauta é todo aquele que permite consultar informação via *Web*.

à Função de Utilidade

Uma Função de Utilidade permite pesar de algum modo as preferências da turma e do Professor.

Uma vez que se pretende que essas preferências seja maximizadas, a função de utilidade deverá ter esse aspecto em consideração.

à Horário

Conjunto de intervalos de tempo que, ao longo da semana de trabalho, podem ou não estar preenchidos com uma determinada "actividade" (no nosso caso, aulas), que têm lugar num determinado local (sala), em que a hora de início e de fim são as indicadas. Além disso, existe um Professor que é o responsável por garantir que essa "actividade" tem lugar. Para uma determinada "actividade" podem existir um ou vários elementos envolvidos (turma(s)).

à JATLite

O JATLite é um pacote de programas em Java que permite uma rápida criação de Agentes e sistemas que comunicam, utilizando a Internet, de forma robusta. Além disso, tem também capacidades específicas de comunicação, sendo a arquitectura dos agentes deixada a cargo do projectista.

à KQML

O KQML é uma linguagem que permite conseguir a comunicação entre os agentes, fornecendo para isso um protocolo que permite o entendimento.

O protocolo é constituído pelos campos:

® *performative*
Indica qual o assunto da mensagem.

® *sender*
Indica quem é o emissor da mensagem.

® *receiver*
Indica quem é o destinatário da mensagem.

® *content*
É um campo opcional que indica o conteúdo da mensagem.

à Preferência

Uma preferência corresponde a um valor que está compreendido entre 0 e 10, correspondendo estes dois valores, respectivamente, a uma preferência mínima e a uma preferência máxima.

É através das preferências que é possível pesar os interesses quer de alunos quer de Professores na geração de um horário.

à Router

O Router é um encaminhador de mensagens, uma vez que todos os agentes enviam/recebem mensagens via Router, sendo da sua responsabilidade dirigir as mensagens para os receptores apropriados, sempre que possível.

Além disso, faz também memorização de mensagens (*message queuing*), ou seja, armazena as mensagens em ficheiros, sendo estas recuperadas e apagadas de acordo com o pedido do agente.

O Router é um ANServer que mantém os endereços dos agentes, os quais podem alterar-se dinamicamente.

Um outro aspecto a referir é que apenas os agentes que se registaram é que podem utilizar o Router.

Por questões de segurança, cada agente tem um nome e uma *password*, a partir dos quais é possível obter acesso ao Router.

Além disso, o Router necessita de conhecer todos os agentes registados, assim como o seu estado de ligação, de tal modo que possa ter uma lista dos agentes.

O Router encarrega-se periodicamente (período de tempo é especificado) de verificar a ligação com cada um dos agentes. Se essas tentativas falharem

um determinado número de vezes (também especificado), o agente é apagado do registo. Isto permite que agentes que se tenham desligado sem enviar mensagem indicativa, continuem a constar da lista de agentes.

à Sistema Multi-Agente

Um Sistema Multi-Agente é um sistema que é desenvolvido com base nas comunicações estabelecidas entre os vários agentes que o integram, tendo cada um objectivos diferentes, ou até mesmo objectivos que permitam a cooperação.

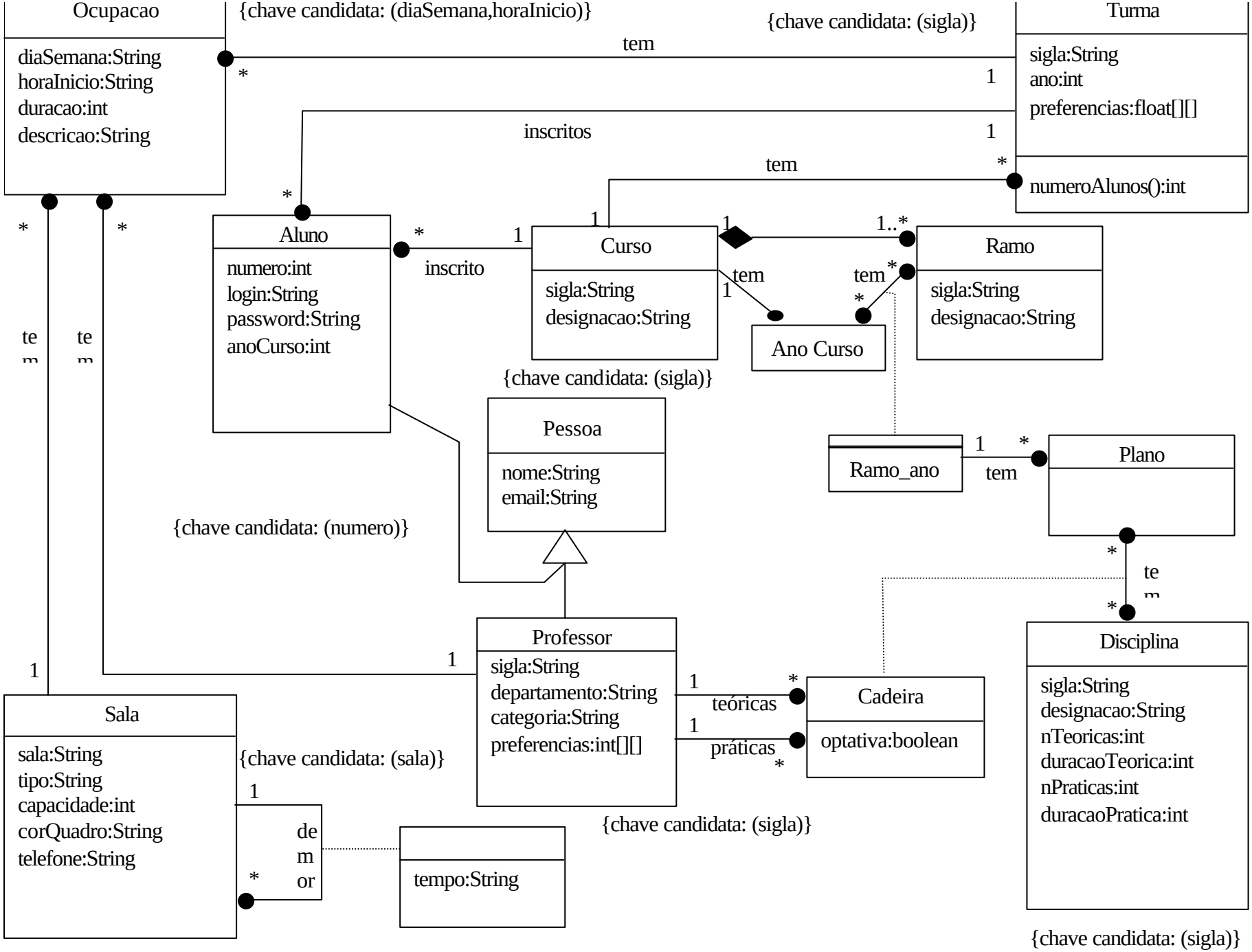
à Turma que inicia a Negociação

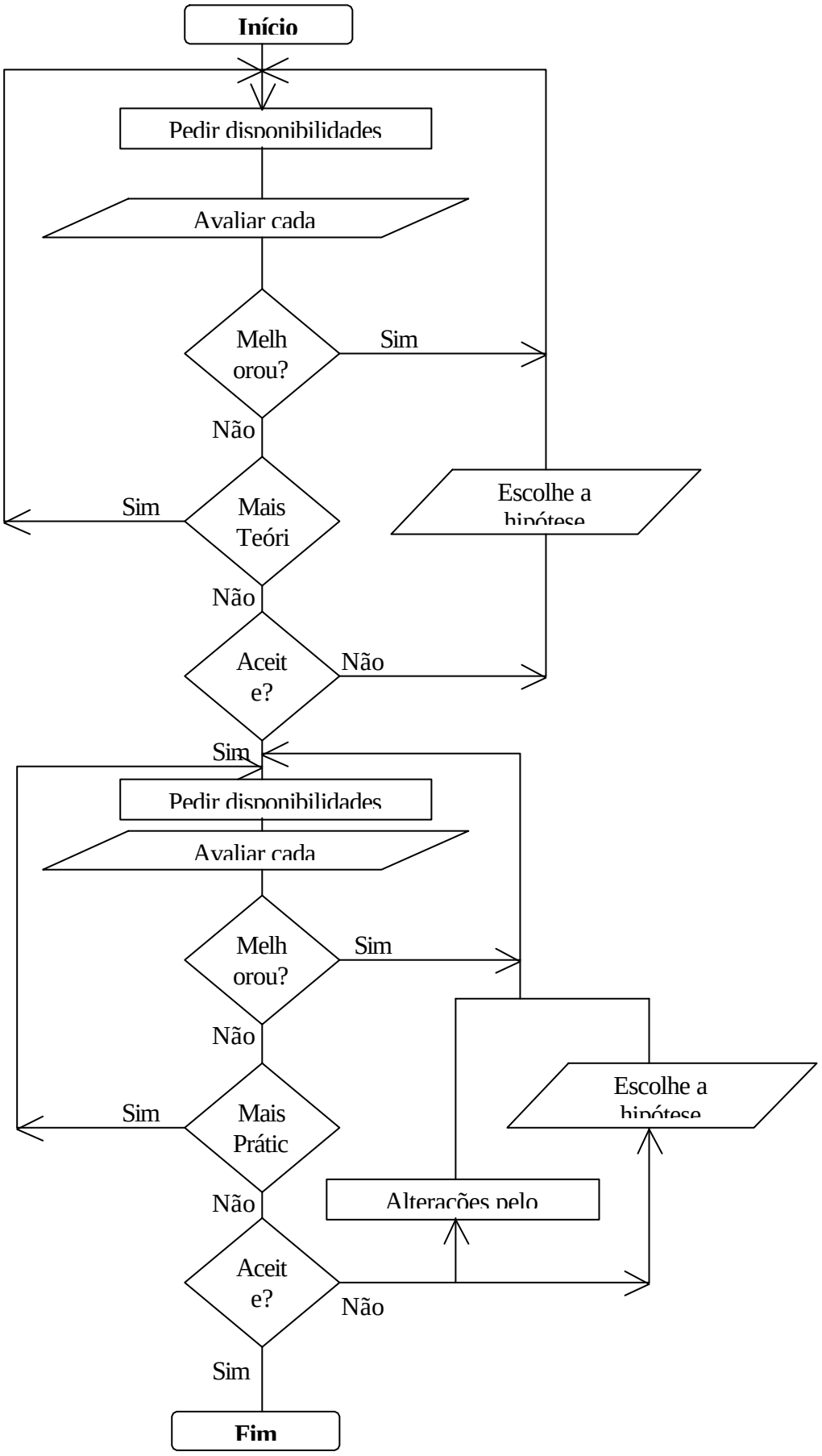
A Turma que inicia a Negociação é um dos agentes deste sistema, que tem como objectivo representar os seus interesses, convencendo outros agentes a aceitar as suas preferências, o que corresponde a maximizar a função de utilidade.

Anexos

Conteúdo:

Diagrama de Classes em UML
Fluxograma para a Geração de Horários


Ocupacao
{chave candidata: (diaSemana,horaInicio)}
diaSemana:String
horaInicio:String
duracao:int
descricao:String
tem
{chave candidata: (sigla)}
Turma
sigla:String
ano:int
preferencias:float[][]
numeroAlunos():int
inscritos
Aluno
numero:int
login:String
password:String
anoCurso:int
inscrito
Curso
sigla:String
designacao:String
{chave candidata: (sigla)}
Ano Curso
Ramo
sigla:String
designacao:String
Pessoa
nome:String
email:String
{chave candidata: (numero)}
Ramo_ano
Plano
Professor
sigla:String
departamento:String
categoria:String
preferencias:int[][]
teóricas
práticas
Cadeira
optativa:boolean
Disciplina
sigla:String
designacao:String
nTeoricas:int
duracaoTeorica:int
nPraticas:int
duracaoPratica:int
{chave candidata: (sigla)}
Sala
sala:String
tipo:String
capacidade:int
corQuadro:String
telefone:String
{chave candidata: (sala)}
dem or
tempo:String


Início
Pedir disponibilidades
Avaliar cada
Melh orou?
Sim
Não
Mais Teóri
Sim
Escolhe a hipótese
Não
Aceit e?
Não
Sim
Pedir disponibilidades
Avaliar cada
Melh orou?
Sim
Não
Mais Prátic
Sim
Escolhe a hipótese
Alterações pelo
Não
Aceit e?
Não
Sim
Fim